# An Energy-Efficient Node Selection Algorithm for Recovering from Faults in the Tree-Based Fog Computing (TBFC) Model

# An Energy-Efficient Node Selection Algorithm for Recovering from Faults in the Tree-Based Fog Computing (TBFC) Model

大間　龍司
Ryuji OMA
指導教員　滝沢　誠

法政大学大学院理工学研究科システム理工学専攻修士課程

In the FC (Fog Computing) model of the IoT (Internet of Things), subprocesses of an application process to handle sensor data are distributed to fog nodes and servers. In the TBFC (Tree-Based Fog Computing) model in our previous studies, fog nodes are hierarchically structured. In this paper, we propose a TBFCG (TBFC for a General process) model to recover from faults of fog nodes. If a node gets faulty, the child nodes are disconnected. We newly propose MET (Minimum Energy in the TBFCG tree) and MPT (selecting Multiple Parents for recovery in the TBFCG tree) algorithms to select new parent nodes for disconnected nodes. A new parent node has to process data from not only the disconnected nodes but also its own child nodes. In the evaluation, the energy consumption and execution time of a new parent node can be reduced by the proposed algorithms.
***Key Words:*** *IoT* (*Internet of Things*), *FC* (*Fog Computing*) model, Energy-efficient FC model, *TBFC model, TBFCG model, MET algorithm, MPT algorithm.*

## 1. INTRODUCTION

In the Internet of Things (IoT), not only computers like servers but also millions of sensors and actuators are interconnected in networks. In the cloud computing model, data collected by sensors is processed by application processes on servers. Networks are congested to transmit the huge volume of sensor data and servers are overloaded to process the sensor data. The fog computing model [1] is proposed to reduce the processing and communication traffic to handle sensor data in the IoT. Sensor data is processed and the output data is sent to another fog node. On receipt of output data, a fog node further processes the data and send the processed data. Thus, servers in clouds finally receive data processed by fog nodes.

In order to reduce the energy consumption and execution time of nodes and servers, the TBFC (Tree-Based Fog Computing) model [2] is proposed. Here, fog nodes are hierarchically structured in a height-balanced tree. Nodes at a root and a bottom levels show root node and edge nodes which communicate with sensors and actuators, respectively. Sensors first send data to edge nodes. Each edge node processes the input data and sends the output data to a parent node. Thus, each node receives data from child nodes and sends processed data to a parent node. Here, the linear model of an application process is considered. Every node at each level of the TBFC tree is equipped with a same subprocess. In order to be tolerant of node failure, the data transmission strategy [3] are proposed.

In this paper, we consider a TBFCG (TBFC for a General process) model where subprocesses of an application process are structured in a tree. Each subprocess is supported by fog nodes. If a node gets faulty, the child nodes are disconnected in the tree. A node which supports the subprocess of the faulty node is an *equivalent* node of the faulty node, which can be a new parent node. We newly propose MET (Minimum Energy in the TBFCG tree) and MPT (selecting Multiple Parents for recovery in the TBFCG tree) algorithms to select new parent nodes in the equivalent nodes for disconnected nodes so that the energy consumption of new parent nodes can be reduced. In the evaluation, we showed the energy consumption of new parent nodes selected in the MET and MPT algorithms.

In section 2, we propose the TBFCG model for a tree-structured application process. In section 3, we discuss how to select an equivalent node of a faulty node. In section 4, we evaluate the MET and MPT algorithms in the TBFCG model.

## 2. TBFCG MODEL

### (1) Tree structure of an application process

In this paper, we consider an application process to handle sensor data, which is hierarchically composed of subprocesses. We consider an example [4] where the TBFCG model is composed of eight nodes as shown in Figure 1. A root node *f* is a server. A pair of sensors $s_1$ and $s_2$ send pairs of temperature

and time data to edge nodes $f_{111}$ and $f_{112}$ and another pair of sensors $s_3$ and $s_4$ send a pairs of humidity and time data to edge nodes $f_{121}$ and $f_{122}$ every one second. A subprocess *tm-aggregate* of the edge nodes $f_{111}$ and $f_{112}$ calculates an average value of temperature data collected for one minute. Another subprocess *hm-aggregate* of the edge nodes $f_{121}$ and $f_{122}$ calculates an average value of humidity data collected for one minute. A parent node $f_{1i}$ receives input data from a pair of child nodes $f_{1i1}$ and $f_{1i2}$ ($i$ = 1, 2). A pair of subprocesses *tm-merge* and *hm-merge* of parent nodes $f_{11}$ and $f_{12}$, respectively, of an edge node $f_{1ij}$ ($i, j$ = 1, 2) sorts and merges multiple temperature and humidity data in time and sends the merged data to its parent node $f_1$. A subprocess *join* of the node $f_1$ joins data of temperature and humidity from the child nodes $f_{11}$ and $f_{12}$. A subprocess *store* of the root node *f* receives joined data and stores the data as a record to the table in the database *DB*. Thus, an application process *P* is a hierarchically composed of subprocesses *tm-aggregate*, *hm-aggregate*, *tm-merge*, *hm-merge*, *join*, and *store* as shown in Figure 1. Figure 2 shows a TBFCG tree of seven fog nodes for the process tree of Figure 1.

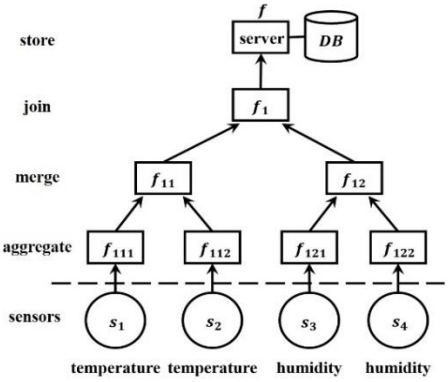


Fig. 1 TBFCG model.

An application process *P* is hierarchically composed of subprocesses. A subprocess *p* is a root node of a process tree. The root subprocess *p* has child subprocesses $p_1, ..., p_c$ ($c \geq 1$) on a root node *f*. Then, a subprocess $p_i$ has child subprocesses $p_{i1}, ..., p_{i,c_i}$ ($c_i \geq 1$). Here, the label *F* of a node $p_F$ shows a path from a root subprocess *p* to the subprocess $p_F$. Each non-root subprocess in process tree *P* is supported by one or more than one fog node. A leaf subprocess $p_F$ is supported by an edge node which communicates with sensors and actuators.

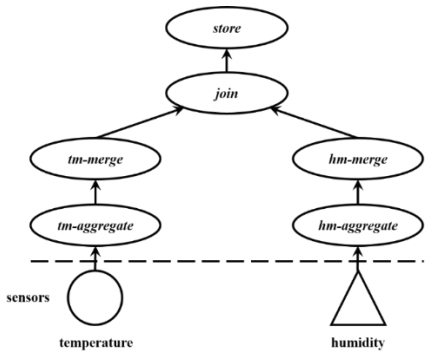


Fig. 2 Process tree.

A node $f_R$ takes a collection $D_R$ of input data $d_{R1}, ..., d_{R,l_R}$ which child nodes $f_{R1}, ..., f_{R,l_R}$ send, respectively. Let $p(f_R)$ be a subprocess supported by a node $f_R$. A subprocess $p(f_R)$ of a node $f_R$ generates output data $d_R$ by processing input data $D_R$. Then, the node $f_R$ sends the output data $d_R$ to a parent node $pt(f_R)$. Finally, data processed by nodes is sent to a root node. A notation $|d|$ shows the size [Byte] of data *d*. The ratio $|d_R|/|D_R|$ is the *output ratio* $\rho_R$ of a node $f_R$.

A pair of nodes $p_R$ and $p_U$ are *equivalent* iff the nodes $f_R$ and $f_U$ support a same subprocess and every pair of ancestor nodes of a same level in $as(f_R)$ and $as(f_U)$ support a same subprocess. In Figure 2, a pair of the nodes $f_{111}$ and $f_{112}$ are equivalent ($f_{111} \equiv f_{112}$) and another pair of the nodes $f_{121}$ and $f_{122}$ are also equivalent ($f_{121} \equiv f_{122}$).

**(2) Execution time of a fog node**

A node $f_R$ takes input data $D_R$ of size $i_R$ (= $|D_R|$) from child nodes and sends output data $d_R$ of size $o_R$ (= $|d_R|$) [2] to a parent node $pt(f_R)$, where $o_R = \rho_R \cdot i_R$ for the output ratio $\rho_R$. A node $f_R$ is realized as a sequence of input ($I_R$), computation ($C_R$), and output ($O_R$) modules. The input module $I_R$ receives input data $D_R$ from the child nodes and the output module $O_R$ sends output data $d_R$ to the parent node $pt(f_R)$. The computation module $C_R$ is a subprocess $p(f_R)$ which generates the output data $d_R$ by processing the input data $D_R$. In this paper, we assume the modules are sequentially performed on receipt of input data.

$TI_R(x)$, $TC_R(x)$, and $TO_R(x)$ show the execution time [sec] of the input $I_R$, computation $C_R$, and output $O_R$ modules of a node $f_R$ for data of size *x*, respectively. $TC_R(x)$ depends on the computation complexity of a subprocess $p(f_R)$ of the node $f_R$. In this paper, $TC_R(x)$ is $ct_R \cdot CM_R(x)$ where $C_R(x) = x$ or $CM_R(x) = x^2$. A pair of execution time $TI_R(x)$ and $TO_R(x)$ to receive and send data of size *x* are proportional to *x*. Here, $ct_R$, $st_R$ and $rt_R$ are constants.

$$TC_R(x) = ct_R \cdot CM_R(x). \quad (1)$$

$$TI_R(x) = rt_R \cdot x. \quad (2)$$

$$TO_R(x) = st_R \cdot x. \quad (3)$$

It takes $TF_R(x)$ [sec] to process input data $D_R$ of size *x* in each node $f_R$:

$$TF_R(x) = TI_R(x) + TC_R(x) + \delta_R \cdot TO_R(\rho_R \cdot x). \quad (4)$$

Here, if $f_R$ is a root, $\delta_R = 0$, else $\delta_R = 1$.

**(3) Energy consumption of a fog node**

$EI_R(x)$, $EC_R(x)$, and $EO_R(x)$ show the electric energy [J] consumed by the input $I_R$, computation $C_R$, and output $O_R$ modules [2] of a node $f_R$ for input data of size *x*, respectively. In this paper, we assume each node $f_R$ follows the SPC (Simple Power Consumption) model [5]. The power consumption of a node $f_R$ to perform the computation module $C_R$ is $maxE_R$ [W]. The energy consumption $EC_R(x)$ [J] of the computation module $C_R$ of a node $f_R$ to process input data of size *x* (> 0) is $EC_R(x) = maxE_R \cdot TC_R(x)$.

A pair of the electric power $PI_R$ and $PO_R$ [W] are consumed by the input $I_R$ and output $O_R$ modules, respectively [5]. $PI_R$ and $PO_R$ are $re_R \cdot maxE_R$ and $se_R \cdot maxE_R$, respectively, where $0 < se_R \leq re_R \leq 1$ in the Raspberry Pi 3 Model B [6] node. A pair of the energy consumption $EI_R(x)$ and $EO_R(x)$ [J] to receive and send data of size *x* (> 0) are $EI_R(x) = PI_R \cdot TI_R(x)$ and $EO_R(x) = PO_R \cdot TO_R(x)$, respectively. Each node $f_R$ consumes the energy $EF_R(x)$ to process input data $D_R$ of size *x*:

$$EF_R(x) = EI_R(x) + EC_R(x) + \delta_R \cdot EO_R(\rho_R \cdot x). \quad (5)$$

## 3. RECOVERY FROM NODE FAULT

A fog node might be faulty in the TBFCG model. If a node $f_R$ gets faulty, every child node $f_{Ri}$ of the node $f_R$ is *disconnected*. No disconnected node can deliver the output data to any ancestor node of the node $f_R$. In the TBFCG model, every pair of nodes at the same level may not support the same subprocess. Only a node equivalent to the faulty node $f_R$ can be a new parent node of disconnected nodes. Let $en(f_R)$ be a set of equivalent nodes of a node $f_R$. We assume each child node $f_{Ri}$ knows every equivalent node of the parent node $f_R$.

Let $f_U$ be an equivalent node of a faulty node $f_R$. Every disconnected child node $f_{Ri}$ is reconnected to a new parent node $f_U$. Here, the node $f_U$ receives data $D_R$ from the nodes $f_{R1}$, ..., $f_{R,l_R}$ in addition to data $D_U$ from its own child nodes $f_{U1}$, ..., $f_{U,l_U}$. Hence, the node $f_U$ has to process both the data $D_U$ and $D_R$ whose total size is $i_U + i_R$ and consumes more energy $EF_U(i_U + i_R)$ than $EF_U(i_U)$. An equivalent node $f_U$ whose energy consumption $EF_U(i_U + i_R)$ is minimum is selected to be a new parent node of all the disconnected nodes. This is the MET (Minimum Energy node in the TBFCG tree) algorithm. However, the energy consumption of the selected a new parent node $f_U$ increases since the node $f_U$ has to process both its own input data $D_U$ and input data $D_R$. In order to reduce the energy consumption and execution time of a new parent node, we propose the MPT (selecting Multiple Parents for recovery in the TBFCG tree) algorithm as follows:

**[MPT algorithm]**

**Input:** $f_R$ = a faulty node;

**Output**: *NP* = a set of new parent nodes;

$F = ch(f_R)$; /* a set of disconnected nodes */

$E = en(f_R)$; /* a set of nodes equivalent to $f_R$ */

**while** $F \neq \emptyset$ **do**

  **select** an equivalent node $f_U$ in $F$ whose size $o_D$ of output data $d_D$ is maximum;

  **select** an equivalent node $f_U$ where $EF_U(i_U + o_D)$ is minimum in $E$;

  **if** $f_U \notin NP$ **then** $NP = NP \cup \{f_U\}$;

  **connect** $f_D$ to $f_U$;

  $ch(f_U) = ch(f_U) \cup \{f_D\}$;

  $pt(f_D) = \{f_U\}$;

  $i_U = i_U + o_D$;

  $F = F - \{f_D\}$;

**while** end;

Here, an equivalent node $f_U$ is selected for each disconnected node $f_{Ri}$. The output data $d_{R1}$, ..., $d_{R,l_R}$ of disconnected nodes are distributed to multiple new parent nodes.

## 4. EVALUATION

We evaluate the MET (Minimum Energy in the TBFCG tree) and MPT (selecting Multiple Parents for recovery in the TBFCG tree) algorithms to select an equivalent node for each disconnected node in the TBFCG (TBFC for a General process) model in terms of the energy consumption of a new parent node. We consider a height-balanced process tree *P* of height *h*. Here, a root subprocess *p* has a pair of child subprocesses $p_1$ and $p_2$. A TBFCG tree for the process tree *P* is a height-balanced four-ary tree with height $h$ ($\geq 1$), where each non-edge node $f_R$ has four child nodes $f_{R1}$, ..., $f_{Rl}$ ($l$ = 4) and every edge node is at level $h - 1$. The root node *f* supports the root subprocess *p* and has four child nodes, $f_1$, $f_2$, $f_3$, and $f_4$. A pair of the child nodes $f_1$ and $f_2$ support the subprocess $p_1$ and another pair of nodes $f_3$ and $f_4$ support the subprocess $p_2$. Then, the child nodes $f_{11}$, ..., $f_{14}$ and $f_{21}$, ..., $f_{24}$ of the nodes $f_1$ and $f_2$, respectively, support the subprocess $f_{11}$ as shown in Figure 3. The child nodes $f_{31}$, ..., $f_{34}$ and $f_{41}$, ..., $f_{44}$ of the nodes $f_3$ and $f_4$, respectively, support the subprocess $p_{21}$. Thus, the subprocesses in the process tree *P* are supported by the nodes in the TBFCG tree. There are totally $4^{h-1}$ edge nodes in the TBFCG tree.

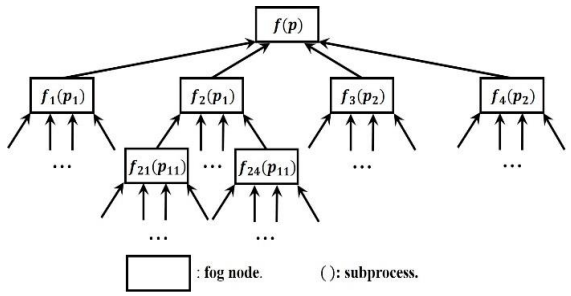


Fig. 3 TBFCG tree.

In the evaluation, we consider two types of data. There are two types of edge nodes, each of which handles one of the two types of data. In the root node *f*, the two types of data are joined. The nodes $f_{1...}$ and $f_{2...}$ handle one type of the data and the nodes $f_{3...}$ and $f_{4...}$ handle the other type of the data. We assume the total size of sensor data is 1 [MB]. The size of sensor data which each edge node receives is randomly decided. The size of sensor data which each edge node receives is randomly decided.

In the evaluation, we assume one node $f_R$ is randomly selected to be faulty at level $l$ ($1 \leq l < h - 1$) and nodes at levels 0 and $h - 1$ are not faulty for simplicity.

We consider the RD1 (Random), RD2, MET, and MPT algorithms. In the RD1 algorithm, an equivalent node is randomly selected to be a new parent node for all the disconnected nodes. On the other hand, in the RD2 algorithm, an equivalent node of a faulty node $f_R$ is randomly selected to be a new parent node for each disconnected node.

Each fog node $f_R$ is assumed to be realized by a Raspberry Pi 3 Model B [6]. Here, the maximum electric power consumption $maxE_R$ is 3.7 [W]. In this paper, we assume a pair of the electric power ratios $re_R$ and $se_R$ of a node $f_R$ are 0.729 and 0.676, respectively, and the execution time ratios $rt_R$, $st_R$, and $ct_R$ are 1, 0.222, and 1, respectively. Hence, a pair of the power $PI_R$ and $PO_R$ of a node $f_R$ are $0.729 \cdot 3.7 = 2.7$ [W] and $0.676 \cdot 3.7 = 2.5$ [W], respectively.

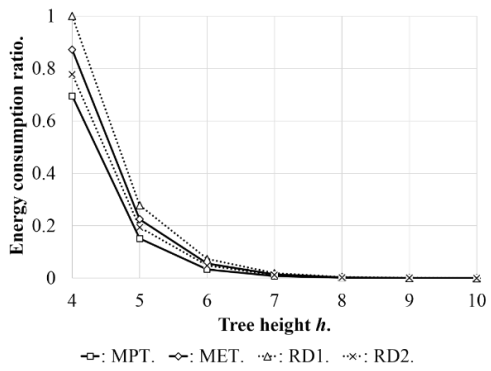


Fig. 4 Energy consumption ratio of new parent nodes for $O(x)$.

of the new parent node selected by the MPT, MET, RD1, and RD2 algorithms for computation complexity $O(x)$ in the GTBFC tree. The energy consumption of new parent nodes selected by the MPT algorithm decreases by about 20 %, 31 %, and 11 % in the MET, RD1 and RD2 algorithms, respectively.

## 5. CONCLUDING REMARKS

In this paper, we proposed the TBFCG model where an application process is hierarchically structured to efficiently realize the IoT. Each subprocess in the process tree is supported by fog nodes. In the TBFCG tree, every pair of nodes at the same level may not support a same subprocess. If a node is faulty, disconnected nodes have to be connected to an equivalent node of the faulty node. In this paper, we proposed the MET and MPT algorithms to select equivalent nodes for a disconnected node. In the evaluation, we showed the energy consumption of equivalent nodes can be reduced in the MPT algorithm compared with the MET, RD1, and RD2 algorithms.


### ACKNOWLEDGEMENT

We would like to thank Prof. Makoto Takizawa for supervising our thesis.


### REFERENCES


1) A. Rahmani, P. Liljeberg, J-S. Preden, A. Jantsch: A Fog Computing in the Internet of Things. Springer, 2018.
2) R. Oma, S. Nakamura, D. Duolikun, T. Enokido, M. Takizawa, "An Energy-Efficient Model for Fog Computing in the Internet of Things (IoT)," *Internet of Things*, Vol.1&2, pp. 14–26, 2018.
3) R. Oma, S. Nakamura, D. Duolikun, T. Enokido, M. Takizawa, "A Fault-Tolerant Tree-Based Fog Computing Model," *Int. J. Web and Grid Services (IJWGS)*, Vol.15, No.3, pp. 219-239, 2019.
4) R. Chida, Y. Guo, R. Oma, S. Nakamura, T. Enokido, M. Takizawa, "Implementation of Fog Nodes in the Tree-Based Fog Computing (TBFC) Model of the IoT," *Proc. of the 7th International Conference on Emerging Internet, Data and Web Technologies (EIDWT-2019)*, pp. 92–102, 2019.
5) T. Enokido, A. Ailixier, M. Takizawa, "An Extended Simple Power Consumption Model for Selecting a Server to Perform Computation Type Processes in Digital Ecosystems," *IEEE Transactions on Industrial Informatics*, Vol.10, No.2, pp. 1627–1636, 2014.
6) Raspberry Pi 3 Model B, URL: https://www.raspberrypi.org/products/raspberry-pi-3-model-b.